1969 (cont'd)

Phoneme depend on both the preceding and following phonemes. This paper provides some needed contextual and probabilistic data about trigram phonemic sequences of spoken English. Since there are approximately 40^3 such sequences, one must discover and study only the more commonly occurring sequences. To this purpose, three types of tables are presented, viz.,
a. Commonly occurring trigram sequences of the form $/\alpha\beta\gamma/$ for every phoneme $/\beta/$.
b. Commonly occurring sequences $/\alpha\beta\gamma/$ for every pair of phonemes $/\alpha/$ and $/\gamma/$.
c. Commonly occurring word boundary sequences of the form $/-\alpha\beta/$ and $/\alpha\beta-/$ where /-/ represents the silence phoneme.
Entries of the above tables contain examples of usage and probabilities of occurrence for each such sequence.

80. Georgia Sutherland, Heuristic Dendral: A Family of LISP Programs, March

The Heuristic Dendral program for generating explanatory hypotheses in organic chemistry is described as an application of the programming language LISP. The description emphasizes the non-chemical aspects of the program, particularly the "topologist" which generates all tree graphs of a collection of nodes.

81. David Luckham, Refinement Theorems in Resolution Theory, March

The paper discusses some basic refinements of the Resolution Principle which are intended to improve the speed and efficiency of theorem-proving programs based on this rule of inference. It is proved that two of the refinements preserve the logical completeness of the proof procedure when used separately, but not when used in conjunction. The results of some preliminary experiments with the refinements are given. Presented at the IRIA symposium on Automatic Deduction, Versailles, France, December 16-21, 1968.

82. Zohar Manna and Amir Pneuli, Formalization of Properties of Recursively Defined Functions, March

This paper is concerned with the relationship between the convergence, correctness and equivalence of recursively defined functions and the satisfiability (or unsatisfiability) of certain first-order formulas.

83. Roger C. Schank, A Conceptual Dependency Representation for a Computer-Oriented Semantics, March

Machines that may be said to function intelligently must be able to understand questions posed in natural language. Since natural language may be assumed to have an underlying conceptual structure, it is desirable to have the machine structure its own experience, both linguistic and nonlinguistic, in a manner concomitant with the human method for doing

so. Some previous attempts at organizing the machine's data base conceptually are discussed. A conceptually-oriented dependency grammar is posited as an interlingua that may be used as an abstract representation of the underlying conceptual structure. The conceptual dependencies are utilized as the highest level in a stratified system that incorporates language-specific realization rules to map from concepts and their relations, into sentences. In order to generate coherent sentences, a conceptual semantics is posited that limits possible conceptual dependencies to statements about the system's knowledge of the real world. This is done by the creation of semantic files that serve to spell out the defining characteristics of a given concept and enumerate the possibilities for relations with other concepts within the range of conceptual experience. The semantic files are created, in part, from a hierarchical organization of semantic categories. The semantic category is part of the definition of a concept and the information at the nodes dominating the semantic category in the hierarchical tree may be used to fill in the semantic file. It is possible to reverse the realization rules to operate on sentences and produce a conceptual parse. All potential parses are checked with the conceptual semantics in order to eliminate semantic and syntactic ambiguities. The system has been programmed; coherent sentences have been generated and the parser is operable. The entire system is posited as a viable linguistic theory

84. David Canfield Smith, MLISP Users' Manual, January
MLISP is a LISP pre-processor designed to facilitate the writing, use, and understanding of LISP programs. This is accomplished through parentheses reduction, comments, introduction of a more visual flow of control with block structure and mnemonic key words, and language redundancy. In addition, some "meta-constructs" are introduced to increase the power of the language.

85. Pierre Vicens, Aspects of Speech Recognition by Computer, April
This thesis describes techniques and methodology which are useful in achieving close to real-time recognition of speech by computer. To analyze connected speech utterances, any speech recognition system must perform the following processes: preprocessing, segmentation, segment classification, recognition of words, recognition of sentences. We present implemented solutions to each of these problems which achieved accurate recognition in all the trial cases.

1969 (cont'd)

86. Patrick J. Hayes, A Machine-Oriented Formulation of the Extended Functional Calculus, April

The Extended Functional Calculus (EFC), a three-valued predicate calculus intended as a language in which to reason about the results of computations, is described in some detail. A formal semantics is given. A machine-oriented (axiomless) inference system for EFC is then described and its completeness relative to the semantics is proved by the method of Semantic Trees. Finally some remarks are made on efficiency.

87. John McCarthy, and A.I. Project Staff, Project Technical Report, June

Plans and accomplishments of the Stanford Artificial Intellligence Project are reviewed in several areas including:

theory (epistemology and mathematical theory of computation),
visual perception and control (Hand-eye and Cart),
speech recognition by computer,
heuristics in machine learning and automatic deduction,
models of cognitive processes (Heuristic DENDRAL, Language Research, and Higher Mental Functions).

This is an excerpt of a proposal to ARPA.

88. Roger C. Schank, Linguistics from a Conceptual Viewpoint (Aspects of Aspects of a Theory of Syntax), April

Some of the assertions made by Chomsky in Aspects of the Theory of Syntax are considered. In particular, the notion of a 'competence' model in linguistics is criticized. Formal postulates for a conceptually-based linguistic theory are presented.

APPENDIX E

STANFORD ARTIFICIAL INTELLIGENCE LABORATORY

OPERATING NOTES

The SAILON series describes the operation of computer programs and equipment used in the Stanford Artificial Intelligence Laboratory.

1) (Superseded by SAILON 28.2, Appendix I.)

2.1) W. Weiher, "Calcomp Plot Routines", revised September 1968.
Describes use of basic plot routines from either FORTRAN IV or MACRO.

3.1) B. Baumgart, "How to Do It and Summaries of Things", revised March 1969. An introductory summary of system features.

4) (Superseded by SAILON 28.2.)

5) W. Weiher, "Preliminary Description of EDIT 2", January 1967.
Describes a Dectape-oriented teletype editor (obsolescent).

6) J. Sauter, "Stanford PDP-6 Time-Sharing System Documentation", January 1967.
Describes some differences between the Stanford Time-Sharing Monitor and the DEC system, including "Spacewar Mode" for real time programs.

7) (Obsolete)

8) S. Russell, "Recent Additions to FORTRAN Library", March 1967.

9) P. Petit, "Electronic Clock", March 1967.
Electronic clock attached to the system gives time in micro-seconds, seconds, minutes, hours, day, month, and year.

10) (Obsolete)

11) P. Petit, "A Recent Change to the Stanford PDP-6 Hardware," March 1967.
The PDP-6 has been changed so that user programs can do their own I/O to devices numbered 700 and above.

12) through 20) (Obsolete)

21) A. Grayson, "The A-D Converter", June 1967.

21 Addendum 1) E. Panofsky, "A/D Converter Multiplexer Patch Panel and Channel Assignments as of 1/9/69", January 1969.

22) (Obsolete)

23) W. Weiher, "Recent Changes to Various System Programs", July 1967.
Describes some modifications to TECO, MACROX, DDT, and LOADER.

24) S. Russell, "PDP-6 I/O Device Number Summary", August 1967.

25) S. Russell, "The Miscellaneous Outputs," August 1967.
Gives bit assignments for output to hydraulic arm and TV camera positioning.

26.1) P. Petit, "FAIL", revised October 1968.
Describes one-pass assembler that is about five times as fast as MACRO and has a more powerful macro processor.

27) P. Land, "PIP", August 1967.
A general file handler (DEC manual is better.).

28.2) L. Quam, "Stanford LISP 1.6 Manual", revised December 1968.
Describes the LISP interpreter and compiler, the editor ALVINE, and other aspects of this venerated list processing system.

29) W. Weiher, "Preliminary Description of the Display Processor", August 1967.
III display system from the programmer's viewpoint.

30) R. Paul, "Operation of the Image Dissector", November 1967.
Theory and operation of the III image dissector camera.

31) J. Sauter, "Disc Diagnostic", October 1967.
A program to test the Librascope Disk and its interface.

32) S. Russell and R. Gruen, "Aid for On-line Computation", October 1967.
Interactive calculator derived from JOSS (DEC manual is better.).

33) K. Pingle, "A List-Processing System for Picture Processing", October 1967.
A set of macros for creating complex data structures in the hand-eye system.

34) DEC Library, "PDP-6 Time Sharing TECO", October 1967.
Interactive string processing system for text editing.

35) K. Pingle, "Hand-Eye Library File", June 1968.

36) G. Feldman, "Fourier Transform Subroutine", June 1968.
FORTRAN subroutine performs one-dimensional Fast Fourier Transform.

37) S. Russell and L. Earnest, "A.I. Laboratory Users Guide", June 1968.
Orientation and administrative procedures.

37 Supplement 1) J. McCarthy, "A.I. Laboratory Users Guide", June 1968.
Hard-line administration.

38) P. Vicens, "New Speech Hardware", August 1968.
Preprocessor for input to speech recognition systems.

39) J. Sauter and D. Swinehart, "SAVE", August 1968.
Program for saving and restoring a single user's disk files on magnetic tape.

40) (Obsolete)

41) L. Quam, "SMILE at LISP", September 1968.
A package of useful LISP functions.

42) G. Falk, "Vidicon Noise Measurements", September 1968.
Measurements of spatial and temporal noise on Cohu vidicon camera connected to the computer.

43) A. Moorer, "DAEMON - Disk Dump and Restore", September 1968.
Puts all or selected files on magnetic tape.

44) A. Moorer, "FCROX - MACROX to Fail Converter", September 1968.
Converts MACROX programs to FAIL format, with a few annotated exceptions.

45) A. Hearn, "REDUCE Implementation Guide", October 1968.
Describes the procedure for assembling REDUCE (a symbolic computation system) in any LISP system.

46) W. Weiher, "Loader Input Format", October 1968.

47 and 47 Supplement 1) J. Sauter and J. Singer, "Known Programming Differences Between the PDP-6 and PDP-10", November 1968.

48) D. Swinehart, "GOGOL III", December 1968.
An algebraic language remeniscent of ALGOL. Includes variable length strings.

49) A. Hearn, "Service Routines for Standard LISP Users", February 1969.

50) W. Weiher, "STOPGAP", February 1969.
A disk-oriented teletype editor.

51) W. Weiher and B. Baumgart, "RPG, Rapid Program Generation", March 1969.
A set of Time Sharing Monitor commands that simplifies the running of text editors, assemblers, compilers, and the loader.

52) A. Moorer, "System Bootstrapper's Manual", February 1969.
How to bring back the system from various states of disarray.